The new exchange format for Danish GIS data

Version 1.0
revision B

Lars V. Nielsen
GisPro

Preface

XDK is a new XML based exchange format specially suited for Danish circumstances. It implements in large the same conceptual ideas as the DSFL basic format adopted in 1997, and is as such a direct replacement for this format.

This document describes how XDK is built, used and implemented. XDK is governed by the steering commitee behind XDK.dk, who has supreme control over what rules are to be upheld when using this format.

XML – in short

XML is a set of rules that determines how conceptual schemas, or languages, are to be created. XML is made and recommended by W3C, the organization that coordinates Internet development, primarily by creating most of, and recommending all, the standards that is used. Other organizations like OpenGIS, WAPforum, Unicode and others can also create standards, which in turn is sent to W3C for adoptation and recommandation.

But XML is not a single language. XML is the core of a whole herd of specifications and languages, all which make a multitude of applications possible. The fact that almost all are XML based themselves means that the support for XML world wide is enormous.

XDK is based on XML to tap into this huge potential.

Terms in this description

In the following descriptions the word”a tag” is used to describe the XML construct that encapsulates commands between’<’ and ’>’, e.g. <X> that is denoted an "X tag". Tags are well known from HTML. Tags in XML are either open tags, <ABC> ...value... </ABC> or singular, closed tags, <ABC/>. All tags are open unless noted otherwise. Possible parameters to a tag is named attributes, e.g. the attribute KODE in the tag <D KODE=”123”>.

Values are the content between the start and end part of an open tag, e.g. <D KODE=”123”>the value</D>. Attribute values in tags are always given with apostrophes, even is the value is numeric.

The description will use the names objects and features somewhat indiscriminately, but the meaning are the same.

The background of XDK

The background for the creation of XDK is based on the realization that the DSFL format in general, and the DSFL basis format specifically, no longer is the common denominator in Danish GIS data exchange that it should be, and partly because the new XML based GML format, currently on the verge of W3C recommandation, is far too complex and unmanageable for most application needs.

XDK is created to close the gap between the old and well known and the new and up-to-date.

XDK gets the bulk of its terminology from the DSFL format, but implements it using XML. That way it is hoped that the"look&feel" from well known DSFL maintained in a modern wrapping will win support from software providers and users alike, so the usage will become as broad as possible.

The website http://www.xdk.dk will be the focal point for all matters concerning XDK. It will contain all necessary descriptions to support usage of the format. The descriptions on the website is authoritative with respect to all XDK related conditions, and the current steering committee at all times runs and maintains this website.

XDK is meant as a replacement for the DSFL format, but only meant as a supplement for GML. XDK cannot replace GML, and certain applications will undoubtably require GML and not XDK. But XDK will be usable under more or less all the conditions that the DSFL format is used under today. A DSFL-to-XDK translator is being made.

An XSLT based transformation from XDK to GML is always in the works, so XDK will not become a blind alley, merely a stepping stone between DSFL and GML. A reverse transformation, from GML to XDK, is a pretty complicated affair, and there is currently no plans to launch such in connection with XDK.

Anyone that has the urge and skills neccessary is welcome to contribute to the creation and development of as many transformation specifications as possible, both as freeware or commercially. The XDK website will upon received notification of such, maintain a published list to all relevant resources.

General rules

XDK formatted data sets can have either the extension *.XDK, or the more generic *.XML. All XDK data sets that doesn't carry one of these extensions is to be regarded as technically and legally invalid. The extension *.XDK is recommended unless technical limitations require usage of the extension *.XML.

All XDK formatted data sets must obey the general rules for XML data sets. Primarily that require all data sets to be ”wellformed XML”, specifically by obeying the XML definition syntax rules. To ensure this a syntax specification has been built, that in version 1.0 is implemented as DTD (Document Type Definition). This DTD specification is accesable online from the XDK.dk website.

XML is typically based on Unicode, using a 16 bit character set which is standardized for the entire world. This is not an absolute requirement, but strongly recommandable. There are numerous Unicode text tools freely available, and there is also a growing number of XML toolkits with graphic user interfaces. If Unicode isn't applied, the proper encoding schema must be stated in the first line in the <xml> tag.

The two first lines in any XDK file must look like this:

```
<?xml version="1.0"?>
<!DOCTYPE XDK SYSTEM "http://www.xdk.dk/DTD/xdk.dtd">
```

The only exception from this rule is usage of an alternative encoding schema. The reference to the DTD specification must look as given above. Any XDK data set that doesn't refer to this specification is invalid. If for any technical reason, e.g. security reasons, is impossible to access the online edition of the DTD, it may be downloaded as installed on a local network/computer and used. But the above validation rule still applies, so any data set referring to such a DTD copy is invalid with regard to any outside parties.

It's illegal to refer to any other syntax specification in an XDK data set. Similarly the officiell DTD may under no circumstances be changed or distorted in any way. If this happens, it will automatically render all data sets referring to the invalid DTD invalid themselves.

The official DTD for version 1.0 doesn't contain syntax rules for numeric and textual attributte values, which implies that data code and feature code tables are outside this specification. However, it is planned that this will be included in a later version of XDK.

As in DSFL, one data set is implemented as one disk file, and seperate data sets cannot be appended without some editing. This has been chosen to maintain the continuity from the DSFL format.

XDK format specification

An XDK data set is defined by the open main tag <XDK>. This is also referred to as the root element. An XDK data set must contain exactly one XDK tag.

The XDK tag is divided into three mandatory parts, included in the given order:

<H-SEKTION> - header or main information for the data set
<R-SEKTION> - reference information, i.e. accuracy information
<D-SEKTION> - data

Tags in each section must also be given in the shown order unless noted otherwise.

H-SEKTION

The H section contains a number of H tags, modelled after the similar %H tokens in the DSFL format. Not all are mandatory, and the primary tag, H123, is a conceptual merge of three DSFL tokens. The possible tags are given in the below table:

H123	mandatory	Closed tag stating the plane and height system plus coordinate sequence
H1	mandatory	Attributte denoting plane coordinate system. Can only be one of the following values: S34J, S34S, S45B, U32, U33, U32W, U33W, LOK, KP2000J, KP2000S, KP2000B
H2	optional	Attributte denoting the height system. Text string that default to DNNGI if not given.
H3	optional	Attributte denoting coordinate sequence. May contain one of the following values: YX, YXZ, XY, XYZ, NE, NEH, EN, ENH. If not given, XY is used. Note that this value doesn't control the reading of coordinate values, as it does in DSFL formatted data. For a detailed description, please refer to the chapter about coordinate notation. In contrast to DSFL data, it's not relevant to add a height component, as XDK is always 3 dimensionel. However, the 3 dimensional versions of the above values are also legal for compatibility reasons. And EN(H) is added to provide for system preferring this notation in UTM coordinates.
HROT	optional	Definition of angle/rotational system. See detailed description in the later chapter about angle values.
AKSE1	mandatory	The axis that is the origin of the angle measurement. May be one of the following enumerated values N, S, Ø, V. These refer to the 4 compass directions.
AKSE2	mandatory	The axis that marks a quarter circumference in positive direction of the angle measurement. May be one of the following enumerated values N, S, Ø, V. These refer to the 4 compass directions.
ENHED	mandatory	Number of units in a full circle, e.g. 400 in a gon/grad based system.
H9	mandatory	Default value for Z, if not given in data.
H11	mandatory	Data provider company name
H12	optional	Data provider address
H13	optional	Data provider postal code (zip)
H14	optional	Data provider postal district (name)

H15	optional	Data provider phone number
H16	optional	Data provider facsimile number
H41	mandatory	Close tag marking creation time of the data
DATO	mandatory	Date in ISO format: YYYYMMDD
KL	optional	Time of day in format HH:M.M.
H58	mandatory	Closed pseudo tag recognizing its inheritence from the DSFL basic format
UDGAVE	fast	Must contain the value ”Basis-udgave 970901”
H59	mandatory	Closed tag giving the version number of the used code table
VERSION	mandatory	Normally version date in the format YYMMDD or YYYYMMDD

An example:

```
<H-SEKTION>
  <H123 H1="S34S" H2="DNNGI" H3="YXZ"/>
  <H9>-99.000</H9>
  <H11>DSFL</H11>
  <H12>Lindevangs Allé 12</H12>
  <H13>2000</H13>
  <H14>Frederiksberg</H14>
  <H15>31861070</H15>
  <H16>31860252</H16>
  <H41 DATO="19970319"/>
  <H58 UDGAVE="Basis-udgave 970901"/>
  <H59 VERSION="19950601"/>
</H-SEKTION>
```

R-SEKTION

The R section contains a series of RN tags, each of which contains a series of ND tags. These are once again modelled after the equivalent tokens in the DSFL format, %RN and %ND. Not all are mandatory. The possible tags are given in the below table:

RN	mandatory	The encapsulating tag, defining an "accuracy class"
KODE	mandatory	ID of the accuracy class. Numeric.
Encapsulated tags:		
ND1	mandatory	Closed tag stating the manufacturing method
KODE	mandatory	Attribute containing an enumerated code for the manufacturing method. Can be one of the following values: DU, DF, DL, SK, SF, SL, UU, FF, LL.
ND11	mandatory	Standard deviation og plane coordinate value
ND12	optional	Standard deviation of height coordinate
ND2X	mandatory	Manufacturing date for paper map, arial photogrammetric exposure or surveying. Equivalent of the DSFL tokens %ND21 - %ND23
ND32	optional	Picture scale if arial photogrammetry
ND41	mandatory	Manufacturer of the digital data
ND5X	optional	Descriptive text, equivalent of the DSFL tokens %ND51 - %ND59. XDK allows an unlimited number of ND5X tags.

Example:

```
<R-SEKTION>
  <RN KODE="1">
    <ND1 KODE="LL"/>
    <ND11>0.10</ND11>
    <ND12>0.05</ND12>
```



```
  <ND2X>19970304</ND2X>
  <ND41>UKENDT</ND41>
  <ND5X>Data eksempel for data i basis-udgave</ND5X>
 </RN>
</R-SEKTION>
```

D-SEKTION

The D section is divided into a series of KU tags, each KU tag conatining a set of features. The KU tag defines the feature code and accuracy class, which then is constant for all encapsulated features.

XDK has 4 object tags, each defining one of the 4 topologic types in the DSFL format: points, polylines, polygons/regions and texts/non-geografic. The KU tag and each of the 4 geografic tags are described in detail below.

KU tag

A KU tag has two mandatory attributes: KODE and N. These attributes are well-known for all DSFL proficients, and contains the feature code (DSFL: class and subclass) and an accuracy ID, respectively. In DSFL terms the KU tag is aggregation of the double-token %K %U and the %N token.

KU	mandatory	Tag for grouping on features by feature code and accuracy class
KODE	mandatory	Attribute containing the feature code
N	mandatory	Attribute containing the accuracy class id
Encapsulates tags:		
D	Optional	Datav values common for the whole group
DU	Optional *)	Text / non-geografc object
P-SEKTION	Optional *)	Point object
L-SEKTION	Optional *)	Polyline object
F-SEKTION	Optional *)	Polygon/region object

*) Each of these can appear any number of times, however grouped in the given order

Example

```
<D-SEKTION>
 <KU KODE="G4_2" N="0">
 ....
```

The packaging of all geographic objects in this KU tag with mandatory attributes, means that all gegraphic objects are defined to have both a feature code and an accuracy class reference.

It is recommended that an DSFL featurecoding %KG4 %U15 is encoded as KODE=”G4_15”, but this is not enforced in version 1.0. Later, when feature code tables are incorporated into the syntax definition, this notation will most probably be used.

A KU tag first contains a set of common value tags, <D>, followed by a series of tags defining geographic objects. Data tags must be given before the geografic objects, but is not neccessary, as each geografic object may contains its own set of data tags.

Common data definitions

Data tags are given by a D tag. A data tag is open, and has a single attribute: KODE. Data tags encapsulates its value, e.g.: <D KODE="131">Hovedgaden</D>. The attribute KODE is mandatory, and must be numeric.

D		Open tag defining a data field
KODE	Mandatory	Attribute naming the D code, given in DSFL code tables. Numeric.
Encapsulatesr:		
	Mandatory	Data value

Example

```
<D-SEKTION>
 <KU KODE="G4_2" N="0">
 <D KODE="111">491</D>
 <D KODE="112">3741</D>
 ....
```

Texts / non-geografic objects

In DSFL the token %DU is used to specify non-geografic data. Similarly the token %T only defines a text position. But in practice the use of %DU has been depreciated and the token %T has been used as a definition tag for a text element. At the same time it's been somwhat unclear which text value was actually the one to be used in the defined text element. In practice, the last defined data tag has been designated for use.

In a structured environment as XML this is clearlt unacceptable. Thus XDK tightens the syntax on this point, defining an open DU tag, encapsulating the neccessary setup tags. The below table states these:

DU		Text or non-geografic feature
Encapsulates tags:		
D	Optional	Data definition. Se nærmere beskrivelse ovenfor.
VV	Optional	Angle (Vinkel) given as a plane coordinate specific rotation angle. Encapsulates the rotation value.
VK	Optional	Angle given by coordinates. Encapsulates two 2D coordinates
TPOS	Optional	Text position Encapsulates one 2D coordinate
ANKER	Optional	Coordinate placement with respect to the text. Is an integer value 1-9. If not given, the value 5 is used (center of text)
TEKST	Mandatory	The text object's textual element
Encapsulates tag:		
KOORD2D	Mandatory	2-dimensional coordinate of text position (see further description)

Example:

```
<DU>
  <D KODE=”112”>3741</D>
  <VV>200.0</VV>
  <TPOS ANKER="4" TEKST="3741">
   <KOORD2D>
    <E>400</E>
    <N>200</N>
   <KOORD2D>
  </TPOS>
</DU>
```

Note that in XDK there's a difference between the data value 3741 and the textual value”3741”.

XDK allows placing several TPOS tags in the same DU feature. This means that one can define a regular multi-text feature, something the DSFL format doesn't allow.

Example:

```
<DU>
  <TPOS ANKER="7" TEKST=" 345">
    <N>133790.663</N>
    <E>77320.0</E>
  </TPOS>
  <TPOS ANKER="7" TEKST=" 346">
    <N>133798.728</N>
    <E>77322.5</E>
  </TPOS>
</DU>
```

Point objects

A point object consists of pure coordinate sets with possible attached data values and angle.

P-SEKTION		
Encapsulated tags:		
D	Optional	Arbitrary number of data values specific to this point feature
VV	Optional *)	Angle by rotational value (see detailed description)
VK	Optional *)	Angle by twin coordinates (see detailed description)
KOORD	At least one	Coordinate set(s)

*) Only one angle value should be given, either using VV or VK

An example:

```
<P-SEKTION>
  <D KODE=“12345“>12345</D>
  <VV>125.0</VV>
  <KOORD>
    <N>133790.663</N>
    <E>77320.0</E>
  </KOORD>
</P-SEKTION>
```

XDK allows multiple KOORD tags in the same P-SEKTION feature. This as a consequence enabled definition of a proper multi point feature, something the DSFL format doesn't allow. In the Norwegian SOSI format this is called a ”svarm”.

An example:

```
<P-SEKTION>
 <KOORD>
  <N>133790.663</N>
  <E>77320.0</E>
 </KOORD>
 <KOORD>
  <N>133798.728</N>
  <E>77322.5</E>
 </KOORD>
</P-SEKTION>
```

Polyline objects

Polyline, or just line, features contains an arbitrary number of seperate partial elements, each containing an arbitrary line sequences. This results in three levels in a line feature: L-SEKTION, L-DEL and L-SEKVENS.

L-SEKTION		A line feature
Indkapsler tags:		
D	Optional	Data field values
L-DEL	Mindst en	Partial element, equivalent of %L2 in the DSFL format
L-DEL encapsulates tags:		
L-SEKVENS		Single sequence in a partial element. Equivalent of %L3 in the DSFL format.
LTYPE	Mandatory	Line type. may be either R (straight lines), S (splines) or C (circular arc). Interpretation matches exactly the definitions used in the DSFL format.
RADIUS	Optional	Is used with line type C if only two coordinate sets are given. If 3 coordinate sets are give, use of RADIUS is unneccessary. Sign of radius value denotes curvature direction, but the interpretation is tightened somewhat wrt. the DSFL format. See further details under the chapter ”Angle values”.
L-SEKVENS encapsulates tag:		
KOORD	At least two	Coordinate sets (see later description)

An example:

```
<L-SEKTION>
 <L-DEL>
  <L-SEKVENS LTYPE="R">
   <KOORD>
    <N>133808.545</N>
    <E>77632.758</E>
   </KOORD>
   <KOORD>
    <N>133820.382</N>
    <E>77634.843</E>
   </KOORD>
  </L-SEKVENS>
 </L-DEL>
</L-SEKTION>
```

Change of line type in same partial element is achieved in the same manner as in the DSFL format. The DSFL sequence %L1KR-%L3KS-%L3KR will look like this in XDK:

```
<L-SEKVENS LTYPE="R">
  .....
</L-SEKVENS>
<L-SEKVENS LTYPE="S">
  .....
</L-SEKVENS>
<L-SEKVENS LTYPE="R">
  .....
</L-SEKVENS>
```

A notable difference between DSFL and XDK, is that the end coordinate in one sequence is not inherited as starting point in the next sequence. This implies that shared points is given twice, once for each sequence. This is necessary to maintain the structural integrity.

Polygon/region objects

Polygon features contains an arbitrary number of seperate partial elements, each containing an arbitrary line sequences. This results in three levels in a line feature: F-SEKTION, F-DEL and F-SEKVENS.

F-SEKTION		A polygon feature
Encapsulates tags:		
D	Optional	Data field values
F-DEL	Mindst en	Partial element, equivalent of %F2/%F4 in the DSFL format. Each partial element must be geographically closed, i.e. the last point being equal to the first point.
YDERKREDS	Optional	May be either J (yes) or N (no). Marks whether or not the partial polygon is an en enclave (YDERKREDS=J) or an exclave or "hole" (YDERKREDS=N). If not given, it defaults to the value J. The values J and N equals %F2 and %F4 resp. in the DSFL format.
F-DEL encapsulates tags:		
F-SEKVENS		Single sequence in partial element. Equivalent to %F3 in the DSFL format.
FTYPE	Mandatory	Line type. may be either R (straight lines), S (splines) or C (circular arc). Interpretation matches exactly the definitions used in the DSFL format.
RADIUS	Optional	Is used with line type C if only two coordinate sets are given. If 3 coordinate sets are give, use of RADIUS is unneccessary. Sign of radius value denotes curvature direction, but the interpretation is tightened somewhat wrt. the DSFL format. See further details under the chapter ”Angle values”.
F-SEKVENS encapsulates tag:		
KOORD	At least 3	Coordinate sets (see later description)

An example:

```
<F-SEKTION>
 <F-DEL YDERKREDS=”J”>
  <F-SEKVENS LTYPE="R">
   <KOORD>
    <N>1000</N>
    <E>2000</E>
   </KOORD>
```



```
    <KOORD>
      <N>2000</N>
      <E>1000</E>
    </KOORD>
    <KOORD>
      <N>1000</N>
      <E>1000</E>
    </KOORD>
    <KOORD>
      <N>1000</N>
      <E>2000</E>
    </KOORD>
  </L-SEKVENS>
 </L-DEL>
</L-SEKTION>
```

See also the functional equivalent description of polyline features.

General conditions

Character sets in XDK files

As mentioned earlier, all XML data sets should be coded in Unicode, as this eliminates all problems in connection with encoding of national characters. This eliminates the need for a special mechanism in XDK for stating the special three Danish characters, that is given by %H0 in the DSFL format.

Unicode is based on a 16 bit character set, making it neccessary to use special tools to work with the files. But there are many such on the market, from the freely available, thru the cheap shareware products to the advanced, expensive programs.

If it's completely impossible to utilize Unicode, normal 8 bit ASCII files are usable. In this case one needs to describe the specific encoding used in the very first XML tag. In Denmark this will typically be the character set ISO 8859-1 (Western Windows ANSI), leaing the first two lines of the XDK file to look like this:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE XDK SYSTEM "http://www.xdk.dk/DTD/xdk.dtd">
```

Coordinate values

Coordinate values are always 3 dimensional in XDK. But leaving the Z value optional, and using the dummy value given by the H9 tag if not given.

In connection with the TPOS tag a special version of the KOORD tag is used: KOORD2D. In the KOORD2D tag it's only possible to insert plane coordinate values. Otherwise the KOORD2D and KOORD tags are completely equivalent.

A single coordinate value given by an E tag (Easting) or an N tag (Northing) can be given in arbitrary order. The actual interpretation of the E and N values depends on the designated plane coordinate system (the H1 attribute in the H123 tag). A possible H3 attribute in the H123 tag is not controlling the coordinate interpretation. The H3 attributte is only to be used as a presentational preference, in which the value"EN" results in a reverse presentational order in the receiving system than the value"NE". The below table describes the default H3 value.

H1 attribute value	H3 attribute default value
S34J, S34S, S45B	YX (equivalent of NE)
U32, U33, U32W, U33W	NE
KP2000J, KP2000S, KP2000B	NE
LOK	XY

Angle values

Angle values given by a VV tag is given with respect to the used plane coordinate system. The below table describes the interpretation of angle values.

H1 attribute	Angle origin axis	Positive rotation	Number of units
S34J, S34S, S45B	X axis toward West	Clockwise 100 gon = North	400 gon/grads
U32, U33, U32W, U33W	N axis toward North	Clockwise 100 gon = East	400 gon/grads
KP2000J, KP2000S, KP2000B	Y axis toward North	Clockwise 100 gon = East	400 gon/grads
LOK	X axis toward East	Counter-clockwise 90 degrees = North	360 degrees

However, it 's optionally possible in XDK to give an alternative angle rotational system. This is done with an HROT tag. This tag denotes the three above parameters, all mandatory in an HROT tag. The table below gives the default values for the individual H1 attribute values:

H1 attribute	AKSE1	AKSE2	ENHED
S34J, S34S, S45B	V	N	400
U32, U33, U32W, U33W	N	Ø	400
KP2000J, KP2000S, KP2000B	N	Ø	400
LOK	Ø	N	(evt.) 360

Angle values given by VK tags is defined using two coordinate sets (KOORD2D tag), in which the angle is defined as the direction from the first to the second point. This method eliminates the need to use and possibly define a rotational system.

In connection with radius values on circular arcs, the value sign is used to control the curvature direction. In the DSFL format minus is used to designate curvature to the left (counterclockwise), and plus to designate curvature to the right (clockwise). This is unfortunate because the interpretation of left/rightis very plane coordinate system dependent, specifically when using a non-normal rotational system with possible axis swaps and reversals.

In XDK the radius sign describes whether the curvature is along or against the positive direction of angle values. The below table shows the differences for each plane coordinate system.

	Rotation i system	Fortegn i DSFL	XDK-situation
S34J, S34S, S45B	rotationen toward right	plus toward right	Sign the same
U32,U33,U32W,U33W	rotationen toward right	plus toward right	Sign the same
KP2000J,KP2000S,KP2000B	n/a	plus toward right	n/a
LOK	rotation toward left (mathematical system)	minus toward left	Sign switched

Interpretation of codes etc.

Code tags are used in two distinct places, just as is the case in the DSFL format: in the feature coding (%K%U, KU tag) and in data value ID (%D, KODE attribute in D tag). These may be validated with respect to seperate code tables, that are occationally updated. These values are not validated in version 1.0 of XDK.

However, the next version of XDK addresses will address this issue, so a certain disciplin even at this stage is in order. The plan is to use the following notation in the future validation schema.

DSFL	XDK	Format
%KG4 %U123	<KU KODE="G4_123" ...>	text
%D123 værdi	<D KODE="123"> værdi </D>	numeric (integer)
%RN id	<RN KODE="id">	numeric (integer)
%N id	<KU ... N="id">	numeric (integer)

Comments on this issue, also alternative suggestions, are welcome.

Comments in data

The DSFL format contained a special tag for entering comments: %B. This is unneccessary in XML files, and thus in XDK, as comments are standardized using this open tag:
<!—any kind of comment -->

XML commentary tags can be entered anywhere, except in the very top of the file, before the initial processing instructions.

Coordinate system values comapared to DSFL

The below table lists the important H1 and H3 values in both DSFL and XDK, to make it easy to find correpondant values.

%H1	%H3	H1-attribut	H3-attribut
S34J	YX	YX	YX
S34S	YX	YX	YX
S45B	YX	YX	YX
U32	NE	NE/YX	NE
U33	NE	NE/YX	NE
U32W	NE	NE/YX	NE
U33W	NE	NE/YX	NE
n/a	n/a	KP2000J	NE
n/a	n/a	KP2000S	NE
n/a	n/a	KP2000B	NE

LOK	XY eller YX	LOK	(som %H3)

Version 1.0 DTD with commentary

DTD content	Commentary
<?xml version="1.0"?>	The essential header
<!-- XDK specification version 1.0 -->	
<!ELEMENT XDK (H-SEKTION,R-SEKTION,D-SEKTION)>	The root element: XDK
	Header information
<!ELEMENT H-SEKTION (H123,HROT?,H9,H11,H12?,H13?,H14?,H15?,H16?,H41,H58,H59)>	**Tag H-SEKTION**
<!ELEMENT H123 EMPTY>	**Tag H123**
<!ATTLIST H123 H1 (S34J|S34S|S45B|U32|U33|U32W|U33W|LOK|KP2000J|KP2000S|KP2000B) #REQUIRED>	
<!ATTLIST H123 H2 CDATA "DNNGI">	
<!ATTLIST H123 H3 (XY|XYZ|YX|YXZ|NE|NEH|EN|ENH) "">	
<!ELEMENT HROT EMPTY>	**Tag HROT**
<!ATTLIST HROT AKSE1 (N|S|Ø|V) #REQUIRED>	
<!ATTLIST HROT AKSE2 (N|S|Ø|V) #REQUIRED>	
<!ATTLIST HROT ENHED CDATA #REQUIRED>	
<!ELEMENT H9 (#PCDATA)>	**Tag H9**
<!ELEMENT H11 (#PCDATA)>	**Tag H11**
<!ELEMENT H12 (#PCDATA)>	**Tag H12**
<!ELEMENT H13 (#PCDATA)>	**Tag H13**
<!ELEMENT H14 (#PCDATA)>	**Tag H14**
<!ELEMENT H15 (#PCDATA)>	**Tag H15**
<!ELEMENT H16 (#PCDATA)>	**Tag H16**
<!ELEMENT H41 EMPTY>	**Tag H41**
<!ATTLIST H41 DATO CDATA #REQUIRED>	
<!ATTLIST H41 KL CDATA #IMPLIED>	
<!ELEMENT H58 EMPTY>	**Tag H58**
<!ATTLIST H58 UDGAVE CDATA #FIXED "Basis-udgave 970901">	
<!ELEMENT H59 EMPTY>	**Tag H59**
<!ATTLIST H59 VERSION CDATA #REQUIRED>	
	References
<!ELEMENT R-SEKTION (RN*)>	**Tag R-SEKTION**
<!ELEMENT RN (ND1,ND11,ND12?,ND2X,ND32?,ND41,ND5X*)>	**Tag RN**
<!ATTLIST RN KODE CDATA #REQUIRED>	
<!ELEMENT ND1 (#PCDATA)>	**Tag ND1**
<!ATTLIST ND1 KODE (DU|DF|DL|SK|SF|SL|UU|FF|LL) #REQUIRED>	
<!ELEMENT ND11 (#PCDATA)>	**Tag ND11**
<!ELEMENT ND12 (#PCDATA)>	**Tag ND12**
<!ELEMENT ND2X (#PCDATA)>	**Tag ND2X**
<!ELEMENT ND32 (#PCDATA)>	**Tag ND32**
<!ELEMENT ND41 (#PCDATA)>	**Tag ND41**
<!ELEMENT ND5X (#PCDATA)>	**Tag ND5X**
	Data
<!ELEMENT D-SEKTION (KU+)>	**Tag D-SEKTION**
<!ELEMENT KU (D*,(DU|P-SEKTION|L-SEKTION|F-SEKTION)+)>	**Tag KU**
<!ATTLIST KU KODE CDATA #REQUIRED>	
<!ATTLIST KU N CDATA #REQUIRED>	
	General state tags: data fields and angles

<!ELEMENT D (#PCDATA)>	**Tag D**
<!ATTLIST D KODE CDATA #REQUIRED>	
<!ELEMENT VV (#PCDATA)>	**Tag VV**
<!ELEMENT VK (KOORD2D,KOORD2D)>	**Tag VK**
	Text positioning
<!ELEMENT TPOS (KOORD2D)>	**Tag TPOS**
<!ATTLIST TPOS ANKER (1|2|3|4|5|6|7|8|9) "5">	
<!ATTLIST TPOS TEKST CDATA #REQUIRED>	
	Coordinate values
<!ELEMENT KOORD2D ((E,N)|(N,E))>	**Tag KOORD2D**
<!ELEMENT KOORD (((E,N)|(N,E)),Z?)>	**Tag KOORD**
<!ELEMENT E (#PCDATA)>	Easting, i.e. coordinate values along East-West axis
<!ELEMENT N (#PCDATA)>	Northing, i.e. coordinate values along North-South axis
<!ELEMENT Z (#PCDATA)>	Z eller kote
	DU feature
<!ELEMENT DU (D*,(VV|VK)?,TPOS+)>	**Tag DU** + after TPOS enables multi texts
	Point features
<!ELEMENT P-SEKTION (D*,VV?,VK?,KOORD+)>	**Tag P-SEKTION** + after KOORD enables multi point
	Polyline features
<!ELEMENT L-SEKTION (D*,L-DEL+)>	**Tag L-SEKTION**
<!ELEMENT L-DEL (L-SEKVENS+)>	
<!ELEMENT L-SEKVENS (KOORD,KOORD+)>	1 + (1-n) = 2-n coordinate sets, minimum 2
<!ATTLIST L-SEKVENS LTYPE (R|S|C) #REQUIRED>	
<!ATTLIST L-SEKVENS RADIUS CDATA #IMPLIED>	
	Polygon/region features
<!ELEMENT F-SEKTION (D*,F-DEL+)>	**Tag F-SEKTION**
<!ELEMENT F-DEL (F-SEKVENS+)>	
<!ATTLIST F-DEL YDERKREDS (J|N) "J">	
<!ELEMENT F-SEKVENS (KOORD,KOORD,KOORD+)>	1 + 1 + (1-n) = 3-n coordinate sets, minimum 3
<!ATTLIST F-SEKVENS FTYPE (R|S|C) #REQUIRED>	
<!ATTLIST F-SEKVENS RADIUS CDATA #IMPLIED>	

